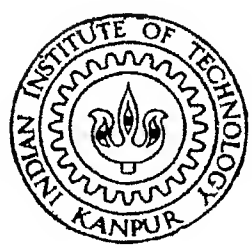
COMPARISON OF FRONT END FEATURES FOR SPEECH RECOGNITION

by
Mangesh Belkhode



DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY KANPUR
FEBRUARY 1998

Comparison of Front End Features for Speech Recognition

A Thesis Submitted
in Partial Fulfillment of the Requirements
for the Degree of
MASTER OF TECHNOLOGY

by

Mangesh Belkhode

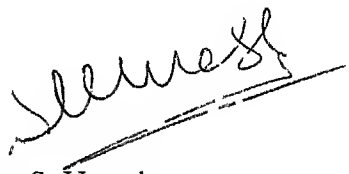
to the

Department of Electrical Engineering
Indian Institute of Technology, Kanpur

February 1998

Certificate

This is to certify that this M Tech thesis work entitled **"Comparison of Front End Features for Speech Recognition** has been carried out by **Mangesh Belkhode** under my supervision and has not been submitted elsewhere for a degree



Dr S Umesh

Assistant professor

Department of Electrical Engineering

Indian Institute of Technology Kanpur

ACKNOWLEDGEMENTS

I expiess my deep gratitude to my thesis supervisor Dr S Umesh for his valuable guidance thioughout my thesis work His encouraging words have always motivated me to woik harder and dispelled the occasional tense moments I thank the faculty at the EE depaitment for enriching my knowledge through their electrifying lectures

I also take the opportunity to express my thanks to my lab mates for their help and support in my thesis work My special thanks to my friend Pushkar for standing by me throughout this endeavour I am also indebted to my friends Hema Parag Shivraj Sameer Pravin and others who have made my stay at IIT Kanpur a memorable one

I cannot but mention the immeasurable love and support I received from my parents In the end I thank the Almighty for making it all possible

ABSTRACT

Considerable success has been achieved in the development of the speaker dependent speech recognition systems The current focus is on the development of the speaker independent speech recognition systems with robustness to noisy environments Obtaining the best parametric representation for the speech signals which is robust to both the above conditions is an important task in designing the speech recognition system This thesis compares the performance of the three front end processors based on LPC mel scale and scale transform in vowel recognition and isolated digit recognition The perfoimance is compared with respect to interspeaker variations and noise For vowels the data is generated from the TIMIT database and for digits it is obtained from Oiegon Giaduate Institute digit database The isolated digit recognition system is based on the vectoi quantization scheme in which multiple reference patterns are used to iepiesent each digit Some earlier studies demonstrated that mel scale based front end piocessois perform better than LPC models and are almost comparable to auditory model based fiont end processois The results in this thesis shows that the scale transform based fiont end piocessors significantly outperform the LPC and the mel scale models under inteispeakei variations and noisy conditions

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

Automatic recognition of speech by machines has been a goal of research for more than four decades and has inspired such science fiction wonders as the robot R2D2 in the George Lucas classic *Star Wars* series of movies By automatic speech recognition we mean a system which takes as input the acoustic waveform produced by the speaker and produces as output a sequence of linguistic words corresponding to the input utterance

However inspite of the glamour of designing an intelligent machine that can recognize the spoken word and comprehend its meaning and inspite of enormous research efforts spent in trying to create such a machine we are far from achieving the desired goal of a machine that can understand spoken words on any subject by all speakers in all environments

In the present scenario we have acceptable speaker dependent speech recognition systems and research is now focused on the development of speaker independent systems By speaker independent speech recognition systems we mean a system which is quite robust to inter speaker variations

There are two main problems for robust speech recognition

1 Differences in the vocal tract size among individual speakers contribute to the variability of speech waveforms The first order effect of a difference in the vocal tract is the scaling of the frequency axis a female speaker for example exhibits

formants roughly 20% higher than the formants from a male speaker with the differences most severe in open vocal tract configurations

2 The inherent mismatch between train and test environment is another problem Speech recognition in adverse conditions has to deal with the effects of different noise levels combined with the influence of different recording channels

Speech recognition has two main approaches

1 The pattern recognition approach The pattern recognition paradigm has four steps namely

1 1 Feature measurement in which a sequence of measurements is made on the input signal to define the pattern

1 2 Pattern training in which one or more patterns corresponding to speech sounds of the same class are used to create a reference pattern of that class

1 3 Pattern classification in which the unknown test pattern is compared with each class reference pattern and a measure of similarity between the test pattern and each reference pattern is computed

1 4 Decision logic in which the overall pattern similarity scores are used to decide which reference pattern best matches the unknown test pattern

2 Hidden Markov Model approach The assumption of this approach is that the speech signal can be well characterized as a parametric random process and that the parameters of the stochastic process can be determined or estimated in a precise well defined manner There are three fundamental problems for HMM design

2 1 The evaluation of probability of sequence of observations given a specific HMM

2 2 The determination of a best sequence of model states

2 3 The adjustment of model parameters so as to best account for the observed signal

Whatever may be the approach used for speech recognition perhaps the greatest common denominator of all speech recognition systems is the signal processing front end which converts the speech signal to some type of parametric representation The selection of the best parametric representation of acoustic data is an important task in the design of

any speech recognition system The usual objectives in selecting a representation are to compress the speech data by eliminating information not pertinent to the phonetic analysis of data and to enhance those aspects of the signal that contribute significantly to the detection of phonetic differences

A wide range of possibilities exist for parametrically representing the speech signal These include the short time energy zero crossing rate level crossing rate and other related parameters Probably the most important parametric representation is the short time spectral envelope Spectral analysis methods are therefore generally considered as the core of the signal processing front end in speech recognition system

Speech recognizers frequently use linear predictive coding (LPC) spectral analysis model or the filter bank spectral analysis model as the front end processors Recently there has been considerable work and interest in using mel cepstral coefficients as acoustic features This is primarily due to the work of Davis and Mermelstein [3] who compared a number of parametric representations and found improved recognition performance using mel based cepstrum A recent article by Jankowski *et al* [4] compares the mel cepstrum with linear prediction and auditory model based front ends They conclude that mel cepstral based front ends significantly outperform LP features and are almost comparable to auditory model based front ends The motivation for using mel frequency based cepstrum stems from psychoacoustic experiments done to study the human auditory system Recently scale cepstrum based front end processor [5] has been proposed as an alternative to both mel cepstral and LPC models for speech spectral analysis The scale transform based cepstrum is motivated by speaker normalization techniques which is necessary since different speakers have different formant frequencies for the same vowel[5]

In this report we have compared the performance of the different signal processing front ends when used for recognition of vowels and isolated digits The report is organized as follows Chapter II describes the basic signal processing front ends In this

chapter some important properties related to linear prediction filterbanks and scale transform are also reviewed Chapter III and IV describes the experimental framework and the procedures for selection of speech data to compare the recognition accuracies of front ends in clean and noisy environments Finally in chapter V the results obtained with various representations are listed and discussed from the point of view of robustness of the front ends with respect to inter speaker variations and noise

2 THE FRONT END PROCESSORS

In this chapter we will discuss the fundamental properties of the linear prediction filterbanks and scale transform Also a complete description of the front end processors is given

2 1 LPC FRONT END PROCESSOR

The theory of linear prediction as applied to speech has been well understood for many years In this section we describe the basics of how LPC has been applied in speech recognition systems The mathematical details and derivations will be omitted here the interested reader is referred to [1] of the Bibliography

2 1 1 The LPC model

The basic idea behind the LPC model is that a given speech signal at time n $s(n)$ can be approximated as a linear combination of the past p samples such that

$$s(n) \approx a_1 s(n-1) + a_2 s(n-2) + \quad + a_p s(n-p) \qquad (2\ 1\ 1)$$

where the coefficients a_1 a_2 a_p are assumed constant over the speech analysis frame We convert Eq (2 1 1) to an equality by including an excitation term $Gu(n)$ giving

$$s(n) = \sum_{i=1}^{p} a\, s(n-i) + Gu(n) \qquad (2\ 1\ 2)$$

where $u(n)$ is a normalized excitation and G is the gain of the excitation By expressing Eq (2 1 2) in the z domain we get the relation

$$S(z) = \sum_{=1}^{p} a\, z \quad S(z) + Gu(z) \tag{2 1 3}$$

leading to the transfer function

$$H(z) = \frac{S(z)}{Gu(z)} = \frac{1}{1 - \sum_{=1}^{p} a\, z^{-}} = \frac{1}{A(z)} \tag{2 1 4}$$

The interpretation of Eq (2 1 4) is given in Figure 2 1 which shows the normalized excitation source $u(n)$ being scaled by gain G and acting as input to the all-pole system $H(z) = \frac{1}{A(z)}$ to produce the speech signal $s(n)$

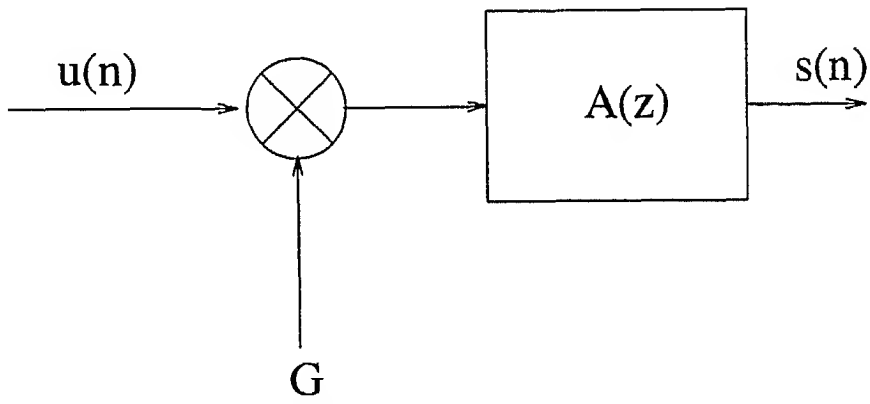


Figure 2 1 LPC model for speech

2 1 2 LPC Analysis Equations

Based on the model of Figure 2 1 the exact relation between $s(n)$ and $u(n)$ is

$$s(n) = \sum_{k=1}^{p} a_k s(n-k) + Gu(n) \tag{2 1 5}$$

We consider the linear combination of past speech samples as the estimate $\tilde{s}(n)$ defined as

$$\tilde{s}(n) = \sum_{k=1}^{p} a_k s(n-k) \qquad (2\ 1\ 6)$$

We now form the prediction error $e(n)$ defined as

$$e(n) = s(n) - \tilde{s}(n) = s(n) - \sum_{k=1}^{p} a_k s(n-k) \qquad (2\ 1\ 7)$$

with error transfer function

$$A(z) = \frac{E(z)}{S(z)} = 1 - \sum_{k=1}^{p} a_k z^{-k} \qquad (2\ 1\ 8)$$

The basic problem of linear prediction analysis is to determine the set of predictor coefficients $\{a_k\}$ directly from the speech signal so that the spectral properties of the digital filter match those of the speech waveform within the analysis window The basic approach is to find a set of predictor coefficients that minimize the mean squared prediction error over a short segment of speech

To set up the equations that must be solved to determine the predictor coefficients we define the short term speech and error segments at time n as

$$\begin{aligned} s_n(m) &= s(n+m) \\ e_n(m) &= e(n+m) \end{aligned} \qquad (2\ 1\ 9)$$

and we seek to minimize the mean squared error at time n

$$E_n = \sum_m e^2(m) \qquad (2\ 1\ 10)$$

which using the definition of $e_n(m)$ in terms of $s_n(m)$ can be written as

$$E_n = \sum_m \left[s_n(m) - \sum_{k=1}^{p} a_k s_n(m-k) \right]^2 \qquad (2\ 1\ 11)$$

To solve Eq (2 1 11) for predictor coefficients we differentiate E_n with respect to each a_k and set the result to zero

$$\frac{\partial E_n}{\partial a_k} = 0 \qquad k = 1\ 2 \quad p \qquad (2\ 1\ 12)$$

giving

$$\phi(i,0) = \sum_{k=1}^{p} \hat{a}_k \phi(i,k) \tag{2.1.13}$$

where

$$\phi(i,k) = \sum_{m} s(m-i)s(m-k) \tag{2.1.14}$$

is the short term covariance of $s(m)$

To solve Eq (2.1.13) for the optimum predictor coefficients we have to compute $\phi(i,k)$ for $1 \le i \le p$ and $0 \le k \le p$ and then solve the resulting set of p equations In practice the method of solving equations is a strong function of the range of m used in defining both the section of speech for analysis and the region over which the mean squared error is computed There are two standard methods of defining this range of speech autocorrelation method and covariance method The covariance method is generally not used for speech recognition system Hence we will not discuss this method but instead will concentrate on the autocorrelation method of LPC analysis for the remainder of the chapter For covariance method the reader may refer to [1]

2.1.3 The Autocorrelation method

A fairly simple and straightforward way of defining the limits on m is to assume that the speech segment is identically zero outside the interval $0 \le m \le N-1$ This is equivalent to assuming that the speech signal $s(n+m)$ is multiplied by a finite window $w(m)$ which is identically zero outside the range $0 \le m \le N-1$ Thus the speech sample for minimization can be expressed as

$$s(m) = \begin{cases} s(m+n)\ w(m) & 0 \le m \le N-1 \\ 0 & otherwise \end{cases} \tag{2.1.15}$$

The effect of the weighting of the speech by a window is illustrated in Figure 2.2 In this figure the upper panel shows the speech waveform the middle panel shows the weighted

section of speech and the bottom panel shows the error signal based on the optimum selection of predictor coefficients

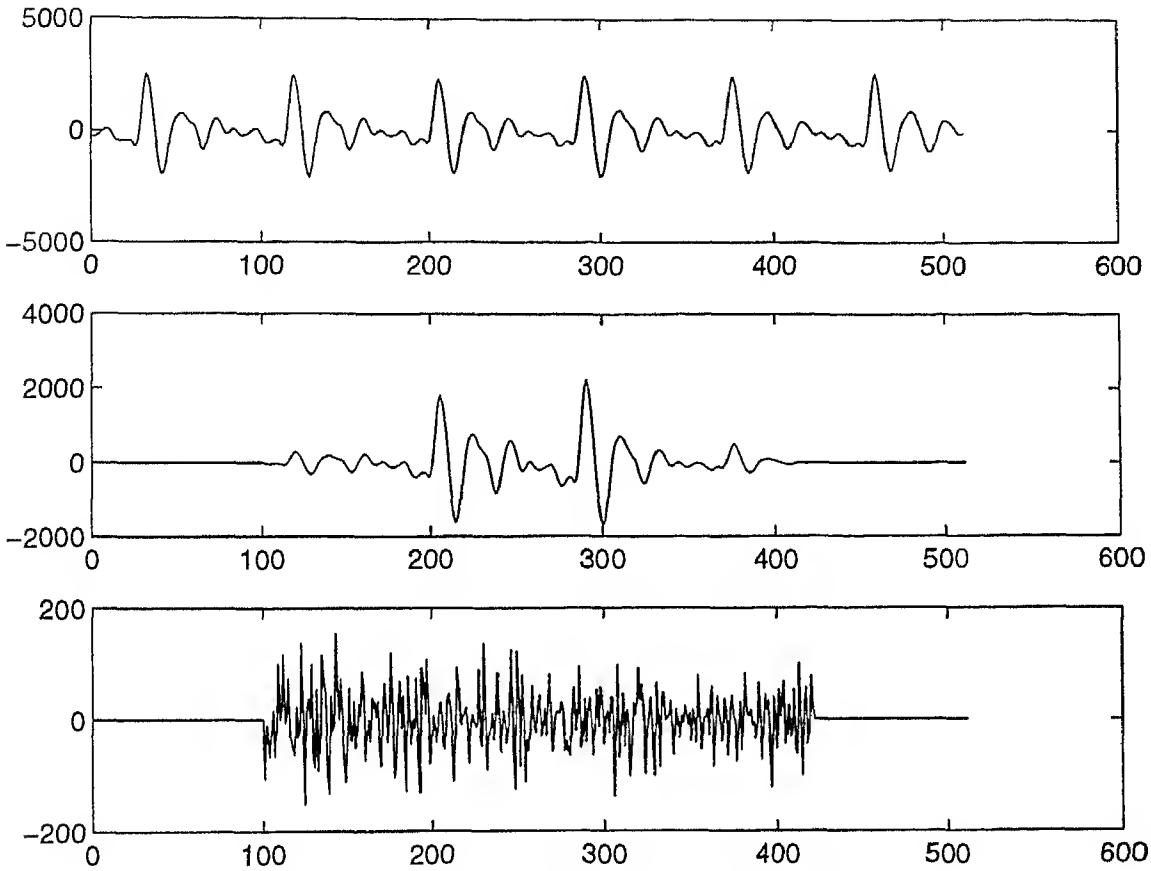


Figure 2 2 The error signal for autocorrelation method

Based on Eq 2 1 15 for $m\langle 0$ the error signal is exactly zero since $s_n(m) = 0$ for all $m\langle 0$ and therefore there is no prediction error Furthermore for $m\rangle N-1+p$ there is no prediction error because $s\ (m) = 0$ for all $m\rangle N-1$ Eq (2 1 10) now become

$$E\ =\ \sum_{m\ 0}^{N\ 1+I} e^2(m) \tag{2 1 16}$$

$$\phi\ (\iota\ k) = \sum_{m\ 0}^{N-1\ (-k)} s\ (m) s_n(m+\iota-k) \qquad 1 \le \iota \le p \qquad 0 \le k \le p \tag{2 1 17}$$

since Eq (2 1 17) is only a function of $i-k$ the covariance function $\phi(i\ k)$ reduces to simple autocorrelation function i e

$$\phi(i\ k)=r(i-k)=\sum_{m=0}^{N-1-(-k)}s(m)s(m+i-k) \qquad (2\ 1\ 18)$$

since the autocorrelation function is symmetric i e $r(-k)=r(k)$ the LPC equations can be expressed as

$$\sum_{k=1}^{p}r(|i-k|)a_k=r(i) \qquad 1\le i\le p \qquad (2\ 1\ 19)$$

and can be expressed in the matrix form as

$$\begin{bmatrix} r(0) & r(1) & r(p-1) \\ r(1) & r(0) & r(p-2) \\ r(p-1) & r(p-2) & r(0) \end{bmatrix}\begin{bmatrix}\hat{a}_1\\ \hat{a}_2\\ \hat{a}_p\end{bmatrix}=\begin{bmatrix} r(1)\\ r(2)\\ r(p)\end{bmatrix} \qquad (2\ 1\ 20)$$

The $p\times p$ matrix of autocorrelation values is a Toeplitz matrix and hence can be solved efficiently through well known procedures One of the commonly used procedure is Levinson Durbin s algorithm and is explained in the next section

2 1 4 LPC processor for speech recognition

Now we describe the details of the LPC front end processor that has been widely used in speech recognition systems Figure 2 3 shows a block diagram of the LPC processor The basic steps in the processing include the following

2 1 4 1 Preemphasis

The digitized speech signal $s(n)$ is put through a low order digital system to spectrally flatten the signal The digital system used is either fixed or slowly adaptive Perhaps the most widely used preemphasis network is fixed first order system

$$H(z)=1-\tilde{a}z^{-1} \qquad 0\ 9\le\tilde{a}\le 1\ 0 \qquad (2\ 1\ 21)$$

In this case the output of the preemphasis network $\tilde{s}(n)$ is related to the input to the network $s(n)$ by the difference equation

$$\tilde{x}_l(n) = x_l(n)w(n) \qquad 0 \le n \le N-1 \tag{2 1 24}$$

A typical window used for autocorrelation method of LPC is the Hamming window which has the form

$$w(n) = 0.54 - 0.46\cos\left(\frac{2\pi n}{N-1}\right) \qquad 0 \le n \le N-1 \tag{2 1 25}$$

2 1 4 4 Autocorrelation analysis

Each frame of windowed signal is next autocorrelated to give

$$r_l(m) = \sum_{n=0}^{N-1-m} \tilde{x}_l(n)\tilde{x}_l(n+m) \qquad m = 0, 1, \ldots, p \tag{2 1 26}$$

where the highest correlation value p is the order of the LPC analysis Typically values of p from 8 to 16 are used in speech recognition systems

2 1 4 5 LPC Analysis

The next processing step is the LPC analysis which converts each frame of $p+1$ coefficients into LPC coefficients The formal method for converting from autocorrelation coefficients to LPC coefficients is known as the Durbin s method and can formally be given as the following algorithm

$$\begin{aligned}
&E^{(0)} = r(0) \\
&k_i = \left\{ r(i) - \sum_{j=1}^{i-1} \alpha_j^{(i-1)} r(|i-j|) \right\} \Big/ E^{(i-1)} \qquad 1 \le i \le p \\
&\alpha_i^{(i)} = k_i \\
&\alpha_j^{(i)} = \alpha_j^{(i-1)} - k_i \alpha_{i-j}^{(i-1)} \\
&E^{(i)} = (1 - k_i^2) E^{(i-1)}
\end{aligned} \tag{2 1 27}$$

The set of equations (2 1 27) are solved recursively for $i = 1, 2, \ldots, p$ and the final solution is given as

$$a_m = \text{LPC coefficients} = \alpha_m^{(p)} \qquad 1 \le m \le p \tag{2 1 28}$$

2 1 4 6 LPC parameter conversion to cepstral coefficients

A very important LPC parameter set which can be derived directly from LPC coefficient set is the LPC cepstral coefficients $c(m)$ The recursion used is

$$c_0 = \ln \sigma$$

$$c_m = a_m + \sum_{k=1}^{m-1} \left(\frac{k}{m}\right) c_k a_{m-k} \qquad 1 \le m \le p \qquad (2.1.29)$$

$$c_m = \sum_{k=1}^{m-1} \left(\frac{k}{m}\right) c_k a_{m-k} \qquad m > p$$

where σ is the gain term in LPC model. The cepstral coefficients have been shown to be more robust, reliable feature set for speech recognition than LPC coefficients[1].

2.2 BANK OF FILTERS FRONT END PROCESSOR

In this section we will first describe how this model reduces the data rate by decomposing the signal into different frequency bands. We will also describe the different types of filter banks that can be used and the various signal processing operations done in the actual filter bank model based front end processor.

2.2.1 Filter bank analysis equations

A block diagram of the canonic structure of a complete filter bank is given in Figure 2.4.

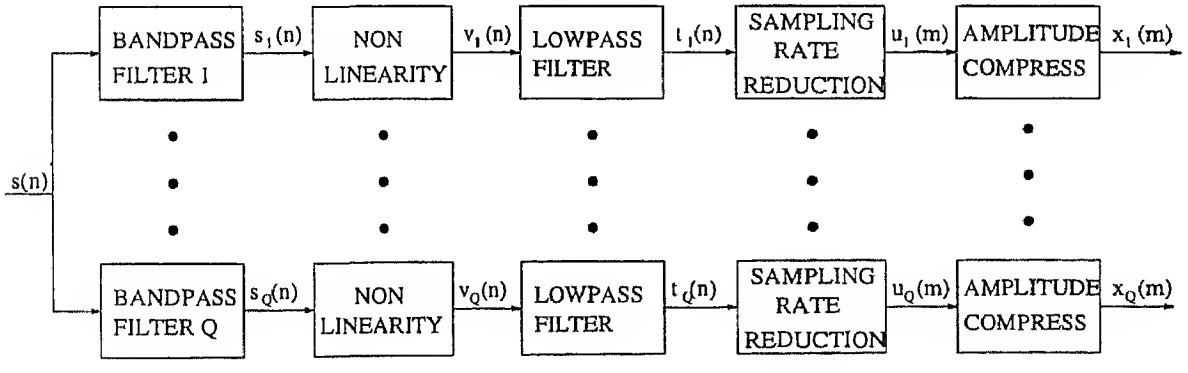


Figure 2.4 Canonic structure of a filter bank

The sampled speech signal $s(n)$ is passed through a bank of Q bandpass filters giving the signals

$$s_i(n) = s(n) * h_i(n) \qquad 1 \le i \le Q \tag{2.2.1a}$$

$$= \sum_{0}^{M-1} h_i(m) s(n-m) \tag{2.2.1b}$$

where we have assumed that the impulse response of the i^{th} bandpass filter is $h_i(m)$ with a duration of M samples hence we use the convolution representation of the filtering operation to give an explicit expression for $s_i(n)$ the bandpass filtered speech signal Since the purpose of the filter bank analyzer is to give a measurement of the energy of the speech signal in a given frequency band each of the bandpass signals is passed through a non linearity such as the full wave or half wave rectifier The non linearity shifts the bandpass signal spectrum to the low frequency band as well as creates high frequency images A low pass filter is used to eliminate the high frequency images giving a set of signals $t_i(n)\ 1 \le i \le Q$ which represents an estimate of the speech signal energy in each of the Q frequency bands The final two blocks of the bank of filter model are a sampling rate reduction block in which the lowpass filtered signals $t_i(n)$ are resampled at the rate on the order of $40-60Hz$ and the signal dynamic range is compressed using an amplitude compression scheme (e g logarithmic encoding)

Now consider the design of a $Q=16$ channel filter bank for a wideband speech signal where the highest frequency signal of interest is $8KHz$ Assume we use a sampling rate of $F_s = 20KHz$ on the speech data to minimize the effects of aliasing in analog to digital conversion The information rate of the raw speech signal is on the order of 240 Kbits per second (20k samples per second times 12 bits per sample) At the output of the analyzer if we use a sampling rate of $50Hz$ and we use a 7 bit logarithmic amplitude compressor we get an information rate of 16 channels times 50 samples per second per channel times 7 bits per sample or 5600 bits per second Thus for this simple example we have achieved about a 40 to 1 reduction in bit rate and hopefully such a data reduction would result in an improved representation of the significant information in the speech signal

2 2 2 Types of filter bank used for speech recognition

The most common type of filter bank used for speech recognition is the uniform filter bank for which the center frequency f of the i^{th} bandpass filter is defined as

$$f_i = \frac{F}{N} i \qquad 1 \le i \le Q \tag{2 2 2}$$

where F is the sampling rate of the speech signal and N is the uniformly spaced filters required to span the frequency range of speech The actual number of filters used in the filter bank Q satisfies the relation

$$Q \le N/2 \tag{2 2 3}$$

with equality when the entire frequency range of the speech signal is used in the analysis The bandwidth b of the i^{th} filter generally satisfies the property

$$b \ge \frac{F}{N} \tag{2 2 4}$$

with equality meaning that there is no frequency overlap and with inequality meaning that adjacent filter channels overlap (If $b < \frac{F}{N}$ then certain portion of the speech spectrum would be missing from the analysis and the resulting speech spectrum would not be considered very meaningful) Figure 2 5 shows a set of Q realistic bandpass filters covering the range from $F/N(1/2)$ to $(F/N)(Q+1/2)$

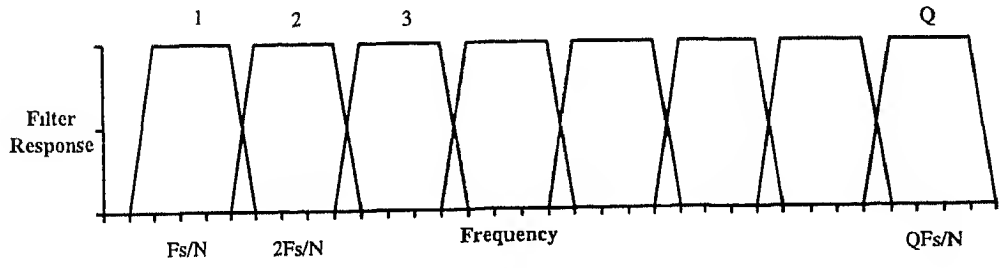


Figure 2 5 Frequency response of a uniform filter bank

The alternative to uniform filterbank is non uniform filterbank designed according to some criterion for how the individual filters should be spaced in frequency One commonly used criterion is to space the filters uniformly along the logarithmic frequency scale which is often justified from a human auditory perception point of view Thus for a set of Q bandpass filters with center frequencies f and bandwidths b $1 \le i \le Q$ we set

$$b_1 = C \qquad (2.2.5a)$$

$$b = \alpha b_{-1} \qquad 2 \le i \le Q \qquad (2.2.5b)$$

$$f = f_1 + \sum_{j=1}^{-1} b_j + \frac{b - b_1}{2} \qquad (2.2.5c)$$

where C and f_1 are the arbitrary bandwidth and center frequency of the first filter and α is the logarithmic growth factor Figure 2.6 shows the set of realistic filters for this filterbank

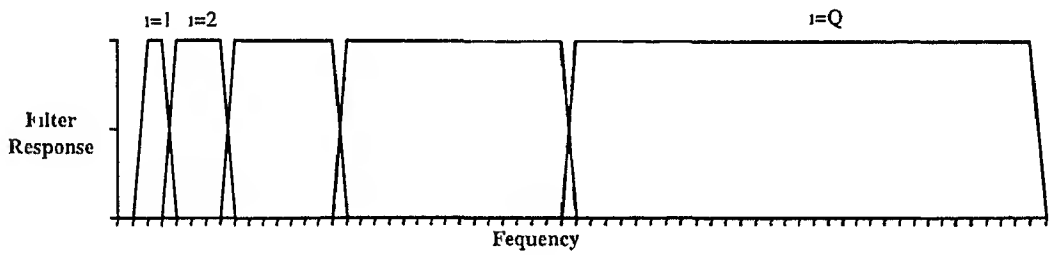


Figure 2.6 Frequency response of a non uniform filter bank

An alternative criterion for designing a non uniform filterbank is to use the critical band scale directly The spacing of the filters along the critical band is based on the perceptual studies and is intended to choose bands that give equal contribution to speech articulation The general shape of the critical band scale is given in Figure 2.7 The scale is close to linear for frequencies below about $1000Hz$ (i.e the bandwidth is essentially a constant as a function of f) and is close to logarithmic for frequencies above $1000Hz$ (i.e the bandwidth is essentially exponential as a function of f)

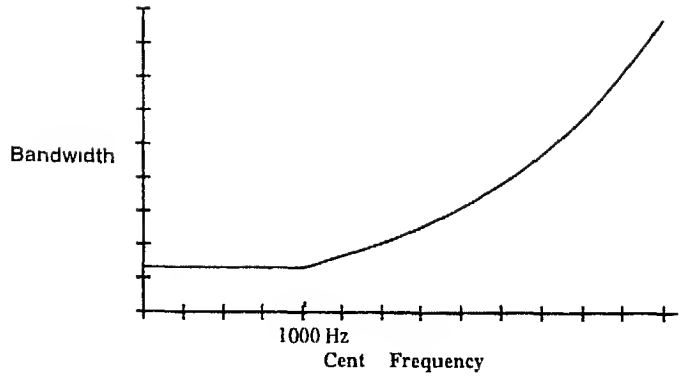


Figure 2 7 Critical band scale

A slight variation of this critical band scale the mel scale is the most widely used and studied filterbank methods For the remainder of this chapter we will concentrate on the mel scale based filterbank model

2 2 3 Mel scale based filterbank model for speech recognition

Figure 2 8 shows a block diagram of this processor

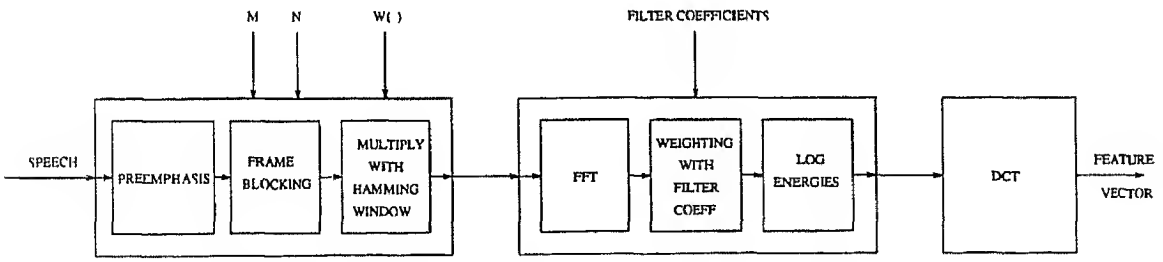


Figure 2 8 Block diagram of a mel scale processor for speech recognition

The signal processing operations involved in this processor are the following

2 2 3 1 Preemphasis frame blocking and windowing

The first three operations are essentially the same as in the LPC processor and has been explained in details in the LPC section earlier

2 2 3 2 Calculation of energy vector

Each of the windowed waveform segment is transformed into the frequency domain by computing the FFT of the corresponding waveform A vector of log energies is then computed from each waveform segment by weighting the FFT coefficients by the magnitude frequency response of the filterbank The log energies are taken for the purpose of dynamic range compression and also in order to make the statistics of the estimated speech power spectrum approximately Gaussian

2 2 3 3 Computing DCT

The final processing stage is to apply the discrete cosine transform (DCT) to the log energy coefficients This has the effect of compressing the spectral information into the lower order coefficients and it also decorrelates them to allow the subsequent statistical modelling to use diagonal covariance matrices

2 3 THE SCALE TRANSFORM BASED FRONT END PROCESSOR

Now we will discuss yet another type of front end processor that has been recently proposed as an alternative to both mel scale and LPC models[5] But before that it is worthwhile to have a look at some of the important properties of the scale transform

2 3 1 The scale transform

The scale transform based cepstrum as it was earlier mentioned is motivated by speaker normalization techniques Such normalization techniques are necessary since different speakers have different formant frequencies for the same vowels A popular procedure for normalization is based on the assumption that the formant values of any given speaker are approximately a multiplicative scale factor times the formant values of any other speaker

for a given vowel In other words the i_{th} formant frequency of two speakers A and B for any vowel are related by

$$F^{(A)} \approx \alpha_{AB} F^{(B)} \tag{2 3 1}$$

where α_{AB} is the scale factor The scale transform[9] is a useful tool to apply in these situations because it provides scale invariant analysis

Now we will briefly review some important properties of the scale transform The scale transform of a function of frequency $X(f)$ is given by

$$D_X(c) = \int_0^\infty X(f) \frac{e^{j\pi c \ln f}}{\sqrt{f}} df \tag{2 3 2}$$

and inversely

$$X(f) = \int_\infty^\infty D_X(c) \frac{e^{+j2\pi c \ln f}}{\sqrt{f}} dc \qquad \forall f > 0 \tag{2 3 3}$$

Now consider the scale transform of $\sqrt{\alpha} X(\alpha f)$ i e

$$D_X^\alpha(c) = \int_0^\infty \sqrt{\alpha} X(\alpha f) \frac{e^{j2\pi c \ln f}}{\sqrt{f}} df \tag{2 3 4}$$

Making the substitution of variables $f' = \alpha f$ we have

$$D_X^\alpha(c) = e^{+j\pi c \ln \alpha} \int_0^\infty X(f') \frac{e^{-j2\pi \ln f'}}{\sqrt{f'}} df' \tag{2 3 5}$$

$$= e^{+j2\pi \ln \alpha} D_X(c) \tag{2 3 6}$$

hence the magnitude of the scale transform of $X(f)$ and its scaled version are the same since the scaling constant α is a part of the phase expression and does not appear in the magnitude of the scale transform It was pointed out in [5] that with respect to the speaker normalization if one were to compute the magnitude of the scale transform of the formant envelope then all speaker dependent scaling constant that appear in the phase term would be removed

The scale transform may also be computed as the Fourier transform of the function $X(e^f)e^{f/2}$ i e

$$D_X(c) = \int_{-\infty}^{\infty} X(e^f)e^{f/2}e^{-j2\pi cf}\,df \qquad (2\ 3\ 7)$$

It may be noted that as a result of log warping i e forming $X(e^f)$ the speaker specific scale constant α is purely a function of the translation parameter in the log warped domain This may be easily seen by considering

$$X_1(f) = X(e^f) \qquad (2\ 3\ 8)$$

$$X_2(f) = X(\alpha e^f) = X(e^{f+\log\alpha}) = X_1(f+\log\alpha) \qquad (2\ 3\ 9)$$

Therefore if there are two formant envelopes that are related by a pure scaling constant that is independent of frequency but is dependent on the pair of speakers then in the log-warped domain the envelopes are the same except for a translation factor dependent on α

In subsequent work [6] it was pointed out that the scale factor is dependent on frequency Based on some experimental results the frequency region of interest is divided into logarithmically equal bands and it is assumed that in each such frequency band the formant envelope of any two speakers are scaled versions of each other In other words the formant envelope of two speakers for the same utterance are assumed to be of the form

$$A(f) = B(\alpha_{AB}^{(i)} f) \qquad f \in i^{th} band \qquad (2\ 3\ 10)$$

and $i = 1\ 2 \quad N$ where N is the number of bands

Further rewriting Eq (2 3 10) as

$$A(f) = B(\alpha_{AB}^{(1+\beta_i)} f) \qquad f \in i^{th} band \qquad (2\ 3\ 11)$$

where

$$\alpha_{AB}^{(i)} = \alpha_{AB}^{(1+\beta_i)} = \alpha_{AB}\ \alpha_{AB}^{\beta_i} \qquad (2\ 3\ 12)$$

Note that α_{AB} is a constant independent of i and is dependent on the pair of speakers while β depends only on the i^{th} frequency band and is independent of pair of speakers

Let $A(f) = B(\alpha_{AB}^{(i)} f)$ be a scaled version of $B(f)$ where $\alpha_{AB}^{(i)}$ is the scaling factor in the i^{th} frequency band If one were to exponentially sample $A(f)$ at Δv spacing in the i^{th} frequency band we have

$$A\left(e^{\Delta v + 1 g(L)}\right) = B\left(e^{\Delta + 1 g(\alpha_A^{(i)}) + 1 g(L)}\right) \qquad m = 0\,1 \qquad M - 1 \tag{2 3 13}$$

where

$$\Delta v = \frac{\log(U) - \log(L)}{M}$$

U L and M are respectively the upper frequency limit lower frequency limit and number of equally spaced samples of the i^{th} band

Rewriting Eq (2 3 13) as

$$A\left(e^{\Delta v + 1 g(L)}\right) = B\left(e^{\left(+\frac{1 g(\alpha_{AB}^{(i)})}{\Delta v}\right)\Delta v + 1 g(L)}\right) \tag{2 3 14}$$

Hence

$$A[m] = B\left[m + \frac{\log(\alpha_{AB}^{(i)})}{\Delta v}\right] \tag{2 3 15}$$

and is translated by $\frac{\log(\alpha_{AB}^{(i)})}{\Delta v_i}$ samples

From Eq (2 3 12) we have

$$\frac{\log(\alpha_{AB}^{(i)})}{\Delta v} = \frac{\log(\alpha_{AB}^{1+\beta})}{\Delta v} = (1+\beta)\frac{\log(\alpha_{AB})}{\Delta v} \tag{2 3 16}$$

If

$$\frac{\Delta v_k}{1+\beta_k} = \frac{\Delta v_j}{1+\beta_j} = \Delta v \tag{2 3 17}$$

then we have equal translation of $\frac{\log(\alpha_{AB})}{\Delta v'}$ in all frequency bands Recalling that we have chosen frequency bands that are equally spaced on logarithmic scale we have

$$\log\left(\frac{U_k}{L_k}\right) = \log\left(\frac{U_j}{L_j}\right) \qquad (2\ 3\ 18)$$

Hence Eq (2 3 17) will be satisfied if

$$(1+\beta_k)M_k = (1+\beta_j)M_j \qquad (2\ 3\ 19)$$

Therefore we use different sampling rates in different frequency bands to achieve warping and the resulting sequences are translated versions of each other For details on computation of β refer to [6]

We have not gone into greater detail here because our goal was to make the reader familiar with the properties of scale transform The interested reader is urged to pursue this fascinating area further by studying the material in [5 6 7]of the Bibliography at the end of this report

2 3 2 Scale transform based front end processor

After a brief introduction of the scale transform we now describe the steps that are involved in the scale transform based front end processor

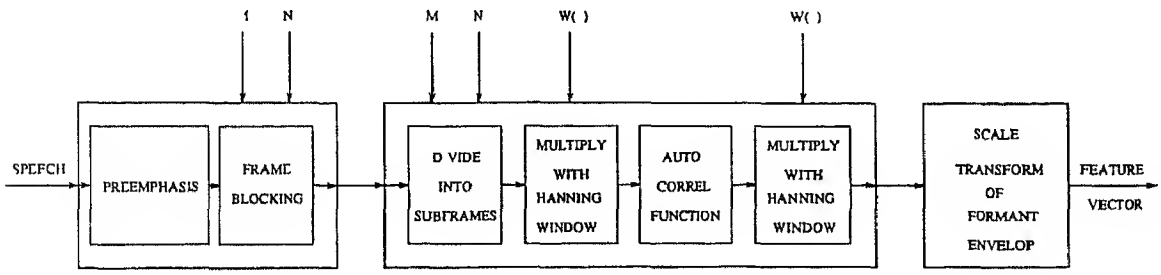


Figure 2 9 Block diagram of a scale cepstrum based front end processor

2 3 2 1 Preemphasis and frame blocking

The first two operations are essentially the same as in the LPC processor and has been explained in details in the LPC section earlier

2 3 2 2 Estimation of the formant envelope

In the spectral domain the speech signal corresponds to the product of the spectrum of the vocal tract filter and the spectrum of the pitch i e

$$V(f) = F(f)P(f) \qquad (2\ 3\ 20)$$

where $V(f)$ $F(f)$ and $P(f)$ are the spectra of the voiced utterance vocal tract and pitch excitation respectively Since we are interested only in the vocal tract response we would like to remove the effects of pitch excitation The following procedure motivated by Nelson s work [17] is used to suppress the effects of pitch Each frame of speech is segmented into Q overlapping subframes and each subframe is Hanning windowed We estimate the sample autocorrelation for each subframe average over the available Q subframes This averaged autocorrelation function is then Hanning windowed and is used to compute the scale cepstrum described in the next section We denote the windowed average autocorrelation estimate as $s[l]$

2 3 2 3 Computing the scale cepstrum

The scale cepstrum is obtained by computing the scale transform of $\log|S(f)|$ and is denoted by $D(c)$ where $S(f)$ is the Fourier transform of $s[l]$ the windowed averaged autocorrelation estimate For computing the scale cepstrum we warp the frequency axis to logarithmic scale i e form $\log|S(e^{f})|$ from $\log|S(f)|$ multiply with preemphasis vector $e^{f/2}$ and compute the Fourier transform to get $D(c)$ The magnitude of the scale cepstrum $|D(c)|$ is then used as a feature vector for speaker independent recognition systems

3 VOWEL CLASSIFICATION

In this chapter we compare the three front end processors by evaluating their performance in vowel classification We also describe the procedure for the selection of data the actual values of the parameters used in the front end processors and the methods for the comparison of the features Finally the results obtained from the classification experiments have been tabulated

3 1 DATABASES

Four different databases RS1 RS2 TS1 and TS2 are used to test the performance of the recognizer based on the three front end processors The data for each database is selected from the dialect region 7 (further divided into dr7train and dr7test) of the TIMIT database The databases consists of the utterances of 10 vowels /aa/ /ae/ /ao/ /ax/ /eh/ /ei/ /ey/ /ih/ /iy/ and /ow/ by male and female speakers Each utterance is so chosen that the corresponding phoneme is relatively stationary over at least 768 samples and the middle 512 samples are used in the computation of the features The noisy utterance is simulated by adding artificially generated white Gaussian noise In our experiments we used clean and noisy speech at 15 db SNR

1 RS1 The data for this database was selected from dr7train and consists of **all** possible utterances of the vowels /aa/ /ae/ /ao/ /ax/ /eh/ /er/ /ey/ /ih/ /iy/ and /ow/ by 20 male speakers and 18 female speakers

2 RS2 The data for this database was also selected from dr7train and consists of 50 utterances of each vowel by male speakers and 50 utterances by female speakers This is done in order to give equal weightage to all vowels

3 TS1 For this database we selected the data from dr7test of dialect region 7 and included **all** possible utterances of ten vowels by 15 male speakers and 8 female speakers

4 TS2 This database is same as TS1 database except that 40 utterances by male speakers and 35 utterances by female speakers were taken for each vowel

3 2 THE VOWEL RECOGNITION SYSTEM

As shown in the Figure 3 1 there are three considerations in implementing a vowel recognition system namely generation of the features creation of the reference patterns and selection of the pattern similarity measure

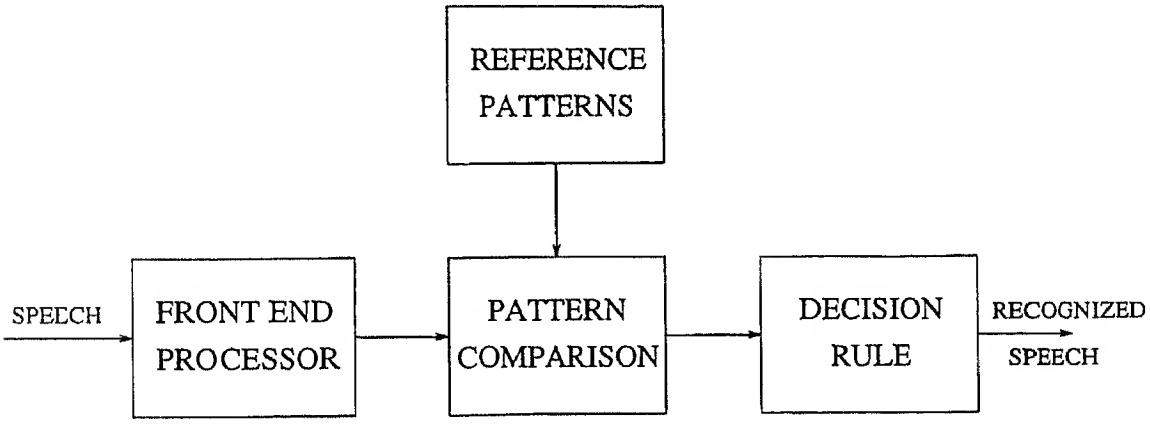


Figure 3 1 Block diagram of a vowel classification system

3 2 1 Generation of the features

The detailed description of the various front end processors is given in chapter 2 Here we give the values of the various parameters used in the processors and describe the procedure for generation of the feature vectors

For obtaining the LPC features the speech waveform is preemphasized by a first order FIR digital filter with $\tilde{a} = 0.97$ We took the frame size as 512 which gives one frame per utterance The frame is weighted with hamming window and $p+1$ autocorrelation values are computed These $p+1$ autocorrelation values are then converted into p LPC coefficients where $p = 9$ These 9 LPC coefficients are converted into 12 LPC cepstral coefficients which are used as the feature vector for vowel recognition

For implementing mel cepstrum the first three steps namely preemphasis frame blocking and weighting with hamming window are same as explained in previous paragraph We then take the Fourier transform of the corresponding waveform to obtain the FFT coefficients A vector of energies is computed by weighting the FFT coefficients with the magnitude response of the filter bank made up of 40 triangular filters The center frequencies of the first 13 linearly spaced filters are 66 67Hz apart starting at 133 34Hz The center frequencies of the other 27 filters are chosen to have a ratio of 1 0711703 between successive filters The mel cepstrum coefficients are then obtained by computing the discrete cosine transform of the vector of log energies

For obtaining the scale cepstrum the first step is preemphasizing and frame blocking as usual The next step is to obtain the smoothed formant envelope which is obtained using the method described in the previous chapter We have chosen the subframes to be 96 samples long and the overlap between the subframes is 64 samples resulting in 13 subframes Since the sampling frequency of the TIMIT database is 16KHz we assume the signal is bandlimited between 100Hz and 7000Hz The scale cepstrum for log warping may therefore be represented as

$$D(c) = \int_{\ln(100)}^{\ln(7000)} \log\left(\left|s\left(e^{v}\right)\right|\right) e^{v/2} e^{-j\,\pi c v}\, dv \qquad (3.2.1)$$

which is the conventional Fourier transform of $\log\left(\left|s\left(e^{\nu}\right)\right|\right)e^{/2}$ For digital implementation we sample in ν domain and obtain an expression which can be easily implemented using the Fast Fourier Transform (FFT) i e (see [5] for details)

$$D\left[\frac{k\,c_l}{N}\right]=\sum_{m=0}^{K-1}\log\left(\left|s\left(e^{\nabla\nu+1\,(100)}\right)\right|\right)e^{\frac{m\nabla\nu+1\,(100)}{2}}e^{-j2\pi\frac{k}{N}m}\qquad k=0\,1\quad(N-1)\qquad(3\,2\,2)$$

where $\nabla\nu=\dfrac{\ln(7000)-\ln(100)}{K-1}$ and $c_p=\dfrac{1}{\nabla\nu}$ The phase term $e^{\frac{j2\pi k\ \ p1\,(100)}{N}}$ can be ignored since it does not contribute to the magnitude of $\left|D\left[\frac{k\,c_p}{N}\right]\right|$ $S\left(e^{m\nabla\nu+1\,(100)}\right)$ can be easily computed from the time lag samples of the smoothed formant envelope $s[l]$ as

$$S\left(e^{\nabla\nu+1\,(100)}\right)=\sum_{-0}^{L-1}s[n]e^{-j2\pi e^{\left|\nabla\ \ln(00)\right|}T}\qquad m=0\,1\quad(K-1)\qquad(3\,2\,3)$$

where T is the sampling period in the time lag domain

Recalling that for making the scale factor α independent of frequency we divide the frequency band between 100Hz and 7000Hz into five logarithmically equal bands of [100 240)Hz [240 550)Hz [550 1280)Hz [1280 3000)Hz and [3000 7000)Hz The scale cepstrum is then computed using the following values of the parameters K=128 L=191 N=256 and $T=\dfrac{1}{16000}$ We have used two warping functions namely warp1 and warp2 In warp1 the number of samples in each of the five frequency bands is given by M_1=9 M_2=12 M_3=21 M_4=35 M_5=51 and in warp2 the values are M_1=40 M_2=32 M_3=23 M_4=18 M_5=15 Note that $D_L[k\ ]$ is interpolated by a factor of two

In all the cepstra the zeroeth coefficient is not used since this is roughly a measure of the spectral energy Coefficients 1 to 12 are used to measure of separability between the different phoneme classes

3 2 2 Forming reference patterns

The next step after the generation of the feature vectors for all utterances is to form the reference patterns for all vowels Let $F_k^{(i)}$ denote the feature vector for k^{th} utterance and N denote the total number of such utterances from the i^{th} phoneme class Then the reference pattern for the i^{th} phoneme class $M^{(i)}$ which is the mean feature vector is given by

$$M^{(i)} = \frac{1}{N}\sum_{k=1}^{N} F_k^{(i)} \qquad i = 1\,2 \quad I \tag{3 2 4}$$

where I is the number of phoneme classes being considered In our case since we are considering 10 vowels I =10

3 2 3 Comparing the patterns

The purpose of this block is to measure the dissimilarity or distance between two features of the speech A test feature vector is compared with each reference feature vector and a distance score is produced In our experiments we used three distance measures

1 Euclidean distance (d_e) The Euclidean distance $d^{(i)}$ between the test vector F and the reference pattern for the i^{th} phoneme class $M^{(i)}$ is given by

$$d_e^{(i)} = \sum_{l=1}^{L}\left(F_l - M_l^{(i)}\right) \tag{3 2 5}$$

where F_l represents the l^{th} coefficient of the feature vector F and $M_l^{(i)}$ represents the l^{th} coefficient in the reference feature vector for the i^{th} phoneme class L is the number of coefficients used For our case L=12

2 Weighted Euclidean distance $(d\ )$ For a test vector F and the reference vector $M^{(i)}$ this distance is given by

$$d^{(\ )} = \sum_{l\ 1}^{I} \frac{\left(F_l - M_l^{(\ )}\right)}{V_l} \qquad (3\ 2\ 6)$$

V_l is the l^{th} coefficient in the mean variance vector V and is given by

$$V_l = \sum_{1}^{I} V_l^{(\ )} \qquad (3\ 2\ 7)$$

where

$$V_l^{(\ )} = \frac{1}{N} \sum_{k=1}^{N} \left(F_{kl}^{(\ )} - M_l^{(\ )}\right)^2 \qquad (3\ 2\ 8)$$

3 Mahalanobis distance using the covariance matrix $(d\ )$ Using this as a measure of dissimilarity between the two patterns the distance between test vector F and the reference pattern $M^{(\ )}$ for the i^{th} phoneme class is given by

$$d^{(\ )} = (F - M^{(\ )})^T C^{-1} \left(F - M^{(\ )}\right) \qquad (3\ 2\ 9)$$

where

$$C = \sum_{-1}^{I} \left[C_{,}\right]^{(\ )} \qquad x = 1\ 2 \quad L \quad y = 1\ 2 \quad L \qquad (3\ 2\ 10)$$

and

$$\left[C_{,}\right]^{(\ )} = \frac{1}{N} \sum_{k=1}^{N} \left(F_{k}^{(\ )} - M^{(\ )}\right)\left(F_{k,}^{(\ )} - M_{,}^{(\ )}\right) \qquad (3\ 2\ 11)$$

$F_k^{(\ )}$ is the x^{th} coefficient in the feature vector for k^{th} utterance of the phoneme class i

3 2 4 Decision rule

The output of the pattern comparison block is a vector $d = d^{(\ )}\ i = 1\ 2 \quad I$ for each utterance The utterance is recognized as phoneme class k if

$$d^{(k)} = \min\left(d^{(\ )}\right) \qquad (3\ 2\ 12)$$

3 3 RESULTS

The results of the recognition tests for databases RS1 RS2 TS1 and TS2 are given in Table 3 1 32 Figure 3 2 and 3 3 shows the recognition accuracy of the system as a function of the distance measure used for the RS1 and TS1 database respectively

		TESTING	
		CLEAN	NOISY
TRAINING	CLEAN	34 54	7 03
	NOISY	20 66	41 58

Table 3 1 LPC Euclidean train RS1 test RS1

		TESTING	
		CLEAN	NOISY
TRAINING	CLEAN	44 41	28 72
	NOISY	25 56	43 27

Table 3 2 Mel Euclidean train RS1 test RS1

		TESTING	
		CLEAN	NOISY
TRAINING	CLEAN	28 80	18 01
	NOISY	13 00	28 28

Table 3 3 Scale(warp1) Euclidean train RS1 test RS1

		TESTING	
		CLEAN	NOISY
TRAINING	CLEAN	35 61	22 61
	NOISY	19 44	31 08

Table 3 4 Scale(warp2) Euclidean train RS1 test RS1

		TESTING	
		CLEAN	NOISY
TRAINING	CLEAN	41 58	19 52
	NOISY	31 57	42 98

Table 3 5 LPC weighted Euclidean train RS1 test RS1

		TESTING	
		CLEAN	NOISY
TRAINING	CLEAN	48 36	30 53
	NOISY	25 71	46 00

Table 3 6 Mel weighted Euclidean train RS1 test RS1

		TESTING	
		CLEAN	NOISY
TRAINING	CLEAN	31 75	15 84
	NOISY	12 82	34 59

Table 3 7 Scale(warp1), weighted Euclidean train RS1 test RS1

		TESTING	
		CLEAN	NOISY
TRAINING	CLEAN	38 01	24 23
	NOISY	20 47	34 84

Table 3 8 Scale(warp2) weighted Euclidean train RS1 test RS1

		TESTING	
		CLEAN	NOISY
TRAINING	CLEAN	45 64	22 62
	NOISY	24 97	44 05

Table 3 9 LPC Mahalanobis train RS1 test RS1

		TESTING	
		CLEAN	NOISY
TRAINING	CLEAN	52 30	44 97
	NOISY	42 84	49 61

Table 3 10 Mel Mahalanobis train RS1 test RS1

		TESTING	
		CLEAN	NOISY
TRAINING	CLEAN	46 22	44 20
	NOISY	36 91	50 72

Table 3 11 Scale(warp1) Mahalanobis train RS1, test RS1

		TESTING	
		CLEAN	NOISY
TRAINING	CLEAN	52 81	45 81
	NOISY	38 34	51 56

Table 3 12 Scale(warp2) Mahalanobis train RS1 test RS1

		TESTING	
		CLEAN	NOISY
TRAINING	CLEAN	33 92	7 73
	NOISY	22 73	40 05

Table 3 13 LPC, Euclidean train RS1 test TS1

		TESTING	
		CLEAN	NOISY
TRAINING	CLEAN	41 90	30 83
	NOISY	22 32	39 10

Table 3 14 Mel Euclidean train RS1 test TS1

		TESTING	
		CLEAN	NOISY
TRAINING	CLEAN	26 13	20 53
	NOISY	11 19	25 89

Table 3 15 Scale(warp1) Euclidean train RS1 test TS1

		TESTING	
		CLEAN	NOISY
TRAINING	CLEAN	31 48	25 41
	NOISY	15 00	27 50

Table 3 16 Scale(warp2) Euclidean train RS1 test TS1

		TESTING	
		CLEAN	NOISY
TRAINING	CLEAN	39 94	20 30
	NOISY	33 37	42 02

Table 3 17 LPC weighted Euclidean train RS1 test TS1

		TESTING	
		CLEAN	NOISY
TRAINING	CLEAN	46 25	32 26
	NOISY	23 63	43 99

Table 3 18 Mel weighted Euclidean train RS1 test TS1

		TESTING	
		CLEAN	NOISY
TRAINING	CLEAN	30 24	19 35
	NOISY	12 26	33 33

Table 3 19 Scale(warp1) weighted Euclidean train RS1 test TS1

		TESTING	
		CLEAN	NOISY
TRAINING	CLEAN	34 40	25 59
	NOISY	18 69	32 14

Table 3 20 Scale(warp2) weighted Euclidean train RS1 test TS1

		TESTING	
		CLEAN	NOISY
TRAINING	CLEAN	42 98	22 56
	NOISY	26 25	43 99

Table 3 21 LPC Mahalanobis train RS1 test TS1

		TESTING	
		CLEAN	NOISY
TRAINING	CLEAN	51 25	44 64
	NOISY	41 67	49 29

Table 3 22 Mel Mahalanobis train RS1 test TS1

		TESTING	
		CLEAN	NOISY
TRAINING	CLEAN	43 87	41 13
	NOISY	40 24	51 55

Table 3 23 Scale(warp1) Mahalanobis train RS1 test TS1

		TESTING	
		CLEAN	NOISY
TRAINING	CLEAN	52 08	43 80
	NOISY	40 77	52 08

Table 3 24 Scalc(warp2) Mahalanobis train RS1 test TS1

		TESTING	
		CLEAN	NOISY
TRAINING	CLEAN	48 80	18 20
	NOISY	22 20	47 80

Table 3 25 LPC Mahalanobis, train RS2 test RS2

		TESTING	
		CLEAN	NOISY
TRAINING	CLEAN	54 70	44 50
	NOISY	42 50	52 70

Table 3 26 Mel Mahalanobis train RS2 test RS2

		TESTING	
		CLEAN	NOISY
TRAINING	CLEAN	49 50	43 80
	NOISY	37 10	55 10

Table 3 27 Scale(warp1), Mahalanobis train RS2 test RS2

		TESTING	
		CLEAN	NOISY
TRAINING	CLEAN	54 60	45 20
	NOISY	38 20	56 00

Table 3 28 Scale(warp2) Mahalanobis train RS2 test RS2

		TESTING	
		CLEAN	NOISY
TRAINING	CLEAN	43 33	18 53
	NOISY	23 87	45 47

Table 3 29 LPC Mahalanobis train RS2 test TS2

		TESTING	
		CLEAN	NOISY
TRAINING	CLEAN	48 93	41 33
	NOISY	36 27	48 93

Table 3 30 Mel Mahalanobis train RS2 test TS2

		TESTING	
		CLEAN	NOISY
TRAINING	CLEAN	40 67	38 00
	NOISY	32 13	50 00

Table 3 31 Scale(warp1) Mahalanobis train RS2 test TS2

		TESTING	
		CLEAN	NOISY
TRAINING	CLEAN	52 00	42 67
	NOISY	36 93	54 00

Table 3 32 Scale(warp2) Mahalanobis train RS2 test TS2

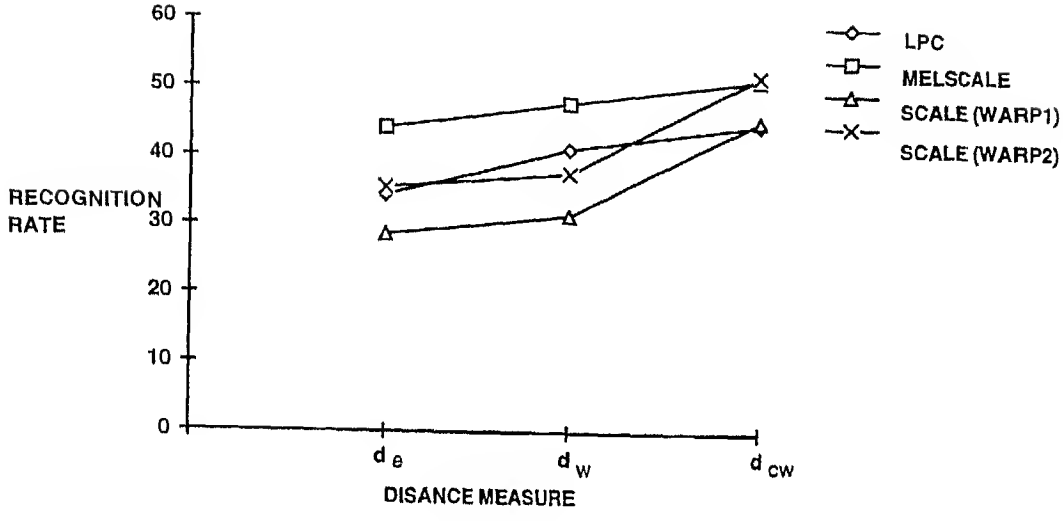


Figure 3 2 Recognition rate vs distance measure clean clean condition RS1 database

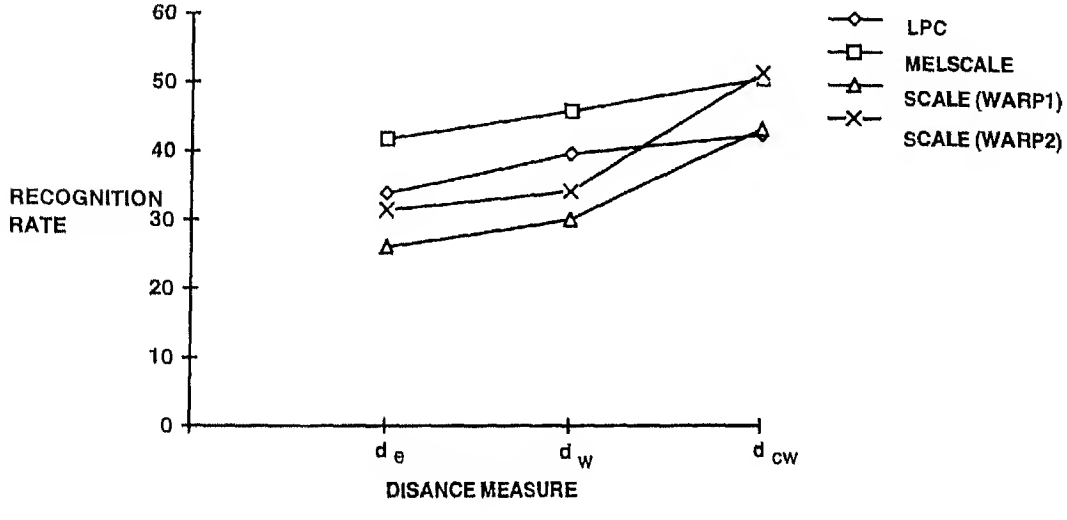


Figure 3 3 Recognition rate vs distance measure clean clean condition TS1 database

Figure 3 4 3 5 shows the performance of the recognition system as a function of signal to noise ratio using different front end processors when Mahalanobis distance is used as a similarity measure The training or generation of reference patterns is done on RS1 database and the testing is done on both RS1 and TS1 databases

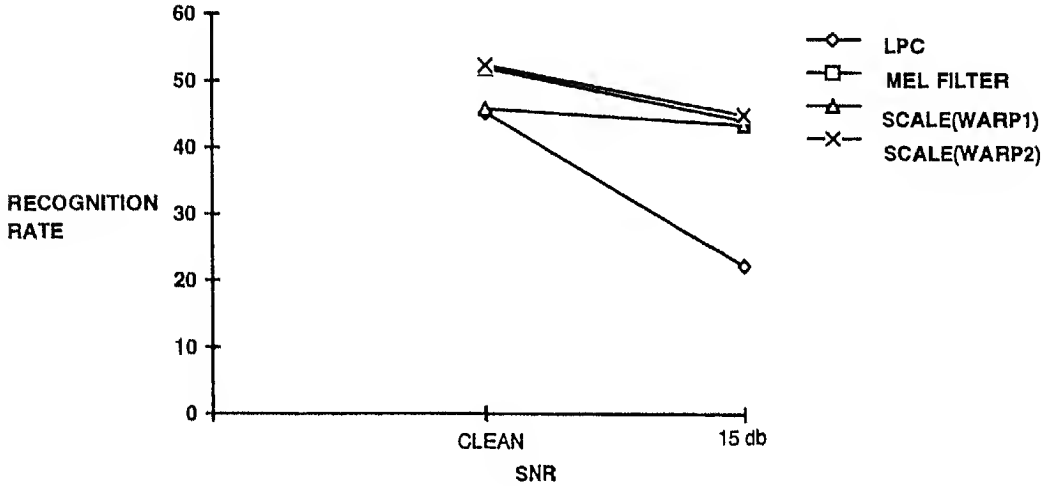


Figure 3 4 RECOGNITION RATE vs SNR TRAIN (CLEAN CONDITION RS1) TEST RS1

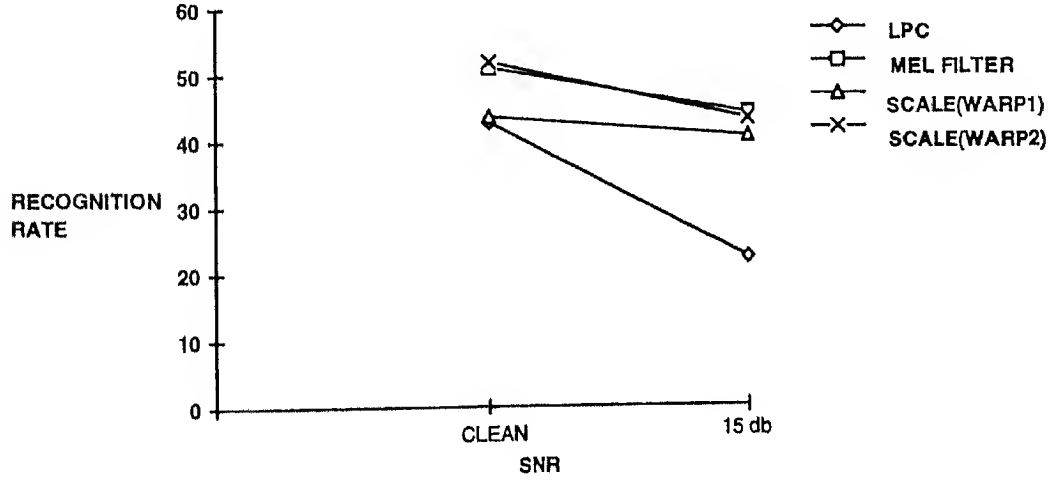


Figure 3 5 RECOGNITION RATE vs SNR TRAIN (CLEAN CONDITION RS1) TEST TS1

The results of Tables 3 1 3 32 show the following

1 The recognition accuracy is a strong function of the distance measure used The accuracy is highest when the distance d_c is used which takes into account the correlation between the cepstral coefficients Since using the distance d_w gives the best performance as compared to the other two henceforth in all our experiments we will use this as a measure of dissimilarity

2 By using d is the distance measure the performance of the scale transform based front end processor enhances significantly as compared to mel scale and LPC front end processor This suggests that the coefficients in the scale cepstrum features are much more correlated as compared to mel scale and LPC features

3 The recognition rates are higher for database RS2 and TS2 as compared to RS1 and TS1 This is probably due to the reason that vowels /iy/ /ih/ /ey/ and /aa/ /ao/ have a very less separability but occur most frequently in speech Vowels /er/ /ow/ /ax/ have a good separability but occur less frequently in speech In database RS2 and TS2 we give equal weightage to all vowels the result being higher recognition rates

4 When the recognition system using LPC features is trained under clean conditions and tested in noisy environments the performance drops significantly as compared to processors using mel scale and scale cepstrum features

5 For vowel recognition the performance of scale (warp1) features is between that of mel based and LPC based features under noisy conditions It can be observed from the tables that there is improved performance of the scale based features using warp2 for vowel classification Under these conditions the vowel classification of the scale based features is comparable to mel based features and better than LPC based features

4 ISOLATED DIGIT RECOGNITION

In this chapter we compare the three front end processors by evaluating their relative recognition performance when used in an isolated digit recognition The isolated digit recognizer is based on a vector quantization scheme The data used in the testing consists of utterances spoken by both male and female speakers to compare the speaker independent recognition performance We also describe the database used actual values of the parameters used in the front end processors the procedures for the formation of the reference patterns and the decision rule used Finally the results obtained with the experiments are tabulated

4 1 DATABASES

We have formed two digit databases RS and TS from the OGI database obtained from Oregon Graduate Institute The speech signal in OGI database is sampled at 8KHz limiting the bandwidth of the speech signal to 4KHz We used the end point detection algorithm of Rabiner et al [16] for all utterances and found that in most cases the entire utterance was included Hence in all our experiments we have included the entire utterance in the analysis

The first database for the digits vocabulary which we call the RS database consisted of a set of 75 speakers (from OGI database) made up of almost equal number of male and female speakers Each talker spoke each digit once giving 10 total utterances per talker and a total of 750 utterances for the RS database The second database TS was also

generated by a set of 75 speakers evenly divided between male and female speakers These 75 talkers were different from the 75 speakers who generated the RS database Every speaker spoke each digit in the vocabulary once resulting in 750 utterances for the TS database

4 2 THE DIGIT RECOGNITION SYSTEM

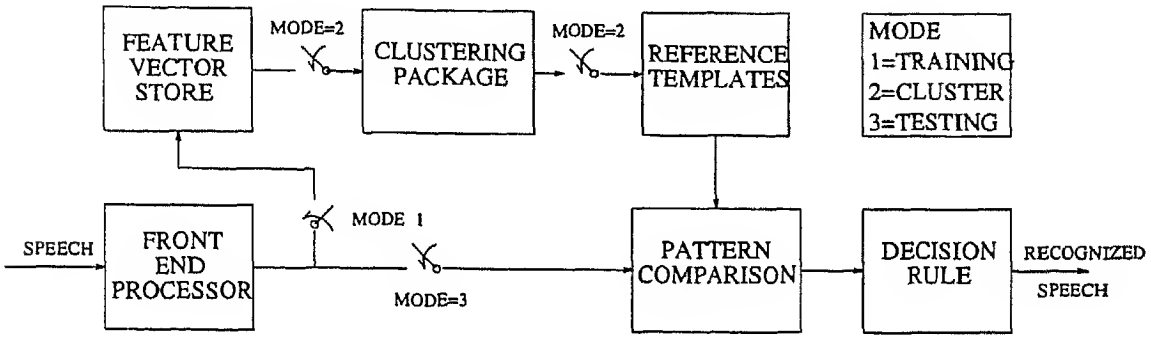


Figure 4 1 Block diagram of an isolated digit recognizer

Figure 4 1 shows the block diagram of the digit recognition system The system operates in three modes namely training template creation or clustering and testing In the training mode the RS database is used to create the feature vectors for each digit and then stored for future use by the clustering package In the second mode the template creation mode the stored data for each digit in the vocabulary is sent to a clustering algorithm which divides it into different clusters The reference patterns are defined to be the center points of the clusters that are found The clustering is performed for all digits of the vocabulary The third mode of the system is the testing or the usage mode in which the unknown test word that is spoken is analyzed and compared to each of the reference templates using a distance measure A vector of distance score is computed for each candidate and finally a decision as to which class it belongs to is taken on the basis of this score

4 2 1 Model The training mode

The procedure for generation of feature vectors with LPC mel scale and scale transform is the same as explained previously except that some values of the parameters are different since we are now using speech sampled at 8KHz and bandlimited to 4KHz Rather than explaining the whole procedure we just enumerate the differences Recall in the previous section each utterance consisted of one frame of 512 samples However each utterance of a digit consists of several phonemes and lasts for a few milliseconds We therefore have to block the utterance into frames and compute the feature vector for each frame Hence corresponding to each utterance of a digit we have a sequence of frame vectors In this section the speech signal is blocked into frames of 320 samples with adjacent frames being separated by 106 samples These correspond to 40msec frames separated by 13 25msec

1 The LPC parameters remain the same as before

2 In the mel scale front end processor firstly we use frames of 320 samples separated by 106 samples and secondly instead of the 40 filters 32 filters are used The first 13 are linearly spaced as usual and other 19 are logarithmically spaced This is done because the spectrum of the speech signal is limited to 4KHz

3 The first difference in the scale transform is the same as above Secondly the frequency region [100 3900]Hz is divided into five bands of [100 240)Hz [240 550)Hz [550 1280)Hz [1280 3000)Hz and [3000 3900]Hz The number of samples in each of the five frequency bands is M_1=9 M_2=12 M_3=21 M_4=35 and M_5=16 for warp1 and the values are M_1=40 M_2=32 M_3=23 M_4=18 and M_5=15 for warp2 It is to be noted that the sampling rate in the fifth frequency band will remain the same as before

4 It has been shown that adding transitional spectral information [14 15] to the instantaneous feature vectors improves the recognition rate significantly Furui [14]

suggested the use of orthogonal polynomials to characterize the time trajectories of cepstral coefficients over a finite length time window to characterize the transitional spectral information The linear regression coefficient namely the first order orthogonal polynomial coefficient is

$$a_m(t) = \left(\sum_{=} F_m(n)\ n \right) \Big/ \left(\sum n^2 \right) \qquad (4\,2\,1)$$

where $F\ (n)$ is the m^{th} coefficient in the n^{th} feature vector The length of the interval is set to 7 frames Accordingly n is equal to 3 The 118msec interval seems adequate for preserving transitional information associated with changes from one phoneme to another The utterance at time t is then represented by cepstrum coefficients $\{F_m(t)\}_{m\,0}^{17}$ and the regression coefficients $\{a_m(t)\}_{m-0}^{12}$ where t is the frame number Since the regression coefficients $a_m(t)$ cannot be defined within the three frame intervals at the beginning and end of speech period these three frame intervals are eliminated from the speech period It doesnot cause the elimination of the actual utterance because of the short silence intervals that are present at the beginning and end of utterances

4 2 2 Mode2 The clustering mode

The basic idea of the clustering is to reduce the information rate of the speech signal to a low rate through the use of a codebook with a relatively small number of codewords or reference patterns The goal is to be able to represent the spectral information of the signal in an efficient manner and yet preserve all the characteristics of the signal The way in which a set of L training vectors can be clustered into a set of M codebook vectors is the following (this procedure is known as generalized Lloyd algorithm or K means clustering algorithm)

1 Initialization Arbitrarily choose M vectors (initially out of the training set of L vectors) as the initial set of codewords in the codebook

2 Nearest neighbour search For each training vector find the codewords in the codebook that is closest and assign that vector to the corresponding cell

3 Centroid update Update the codewords in each cell using the centroid of the training vectors assigned to that cell

4 Iteration Repeat steps 2 and 3 until the average distortion falls below a preset threshold

Although the above iterative procedure works well it has been shown that it is advantageous to design an M vector codebook in stages i e by first designing a 1 vector codebook then using a splitting technique on the codewords to initialize the search for a 2 vector codebook and continuing the splitting process until the desired M vector codebook is obtained This procedure is called the binary split algorithm and is formally implemented by the following procedure

1 Design a 1 vector codebook this is the centroid of the entire set of training vectors

2 Double the size of the codebook by splitting each current codebook y_n according to the rule

$$y^+ = y_n(1+\varepsilon) \qquad (4\,2\,2)$$

$$y^- = y\ (1-\varepsilon) \qquad (4\,2\,3)$$

where n varies from 1 to current size of the codebook and ε is the splitting parameter We use ε =0 01

3 Use the K means iterative algorithm to get the best set of centroids for the split codebook

4 Iterate steps 2 and 3 until a codebook of size M is designed

We tried different values of M and found that a value of M=16 suits the best both from the point of view of the recognition performance and computational complexity Hence each digit is represented by M such codewords

4 2 3 Mode3 The testing mode

The last mode is the testing or the usage mode in which the unknown utterance is compared with each of the codewords in the codebook for all digits In the previous chapter we used three distance measures and from the results it was seen that using d_w as the distance measure gives least error rates Hence in all subsequent experiments we use this as a measure of dissimilarity

Let the feature vector at time $t = q$ be represented by $F(q)$ and $q = 1\ 2 \quad Q$ where Q is the total number of frames when the utterance is blocked into frames of size 320 samples with a separation of 106 samples Let the j^{th} codeword in the codebook for the i^{th} digit be represented by $R_j^{(i)}$ then the distance between $F(q)$ and $R_j^{(i)}$ is represented by $d\left(F(q)\ R_j^{(i)}\right)$ where subscript we has been omitted For all the codewords $j = 1\ 2 \quad M$ where M is the total number of codewords in the codebook we find

$$D^{(i)}(q) = \min_{1 \le j \le M} d\left(F(q)\ R_j^{(i)}\right) \tag{4 2 4}$$

and

$$D^{(i)} = \sum_{q=1}^{Q} D^{(i)}(q) \tag{4 2 5}$$

The value $D^{(i)}$ is calculated for all digits $i = 1\ 2 \quad I$ and the decision rule is the recognized digit is k if

$$D^{(k)} = \min\left(D^{(i)}\right) \tag{4 2 6}$$

where k^{th} codebook is for digit k

4 3 RESULTS

To evaluate the accuracy of the digit recognizer using the different front end processors a series of digit recognition experiments was performed The measure of the performance was the overall recognition accuracy

Three experiments were run on the databases RS and TS The experiments were as follows

1 Experiment no 1 No transitional spectral information was included while performing this experiment Table 4 1 6 tabulates the recognition accuracy of the system using different front end processors Noisy speech at the signal to noise ratio of 15db was used

		TESTING	
		CLEAN	NOISY
TRAINING	CLEAN	92 53	16 00
	NOISY	43 46	92 80

Table 4 1 LPC WTI TRAIN RS TEST RS

		TESTING	
		CLEAN	NOISY
TRAINING	CLEAN	95 60	10 53
	NOISY	10 26	89 73

Table 4 2 MEL WTI TRAIN RS TEST RS

		TESTING	
		CLEAN	NOISY
TRAINING	CLEAN	92 13	30 40
	NOISY	11 33	89 46

Table 4 3 SCALE WTI TRAIN RS TEST RS

		TESTING	
		CLEAN	NOISY
TRAINING	CLEAN	85 46	13 86
	NOISY	36 00	85 06

Table 4 4 LPC WTI TRAIN RS TEST TS

		TESTING	
		CLEAN	NOISY
TRAINING	CLEAN	82 93	10 26
	NOISY	10 13	78 80

Table 4 5 MEL WTI TRAIN RS TEST TS

		TESTING	
		CLEAN	NOISY
TRAINING	CLEAN	82 53	28 00
	NOISY	10 53	79 73

Table 4 6 SCALE WTI TRAIN RS TEST TS

2 Experiment no 2 In this experiment the transitional spectral information was included as suggested by Furui [14] giving a total of 24 coefficients in each feature vector Noisy

speech at the signal to noise ratio of 15db was used Tables 4 7 12 shows the recognition accuracy

		TESTING	
		CLEAN	NOISY
TRAINING	CLEAN	98 00	10 13
	NOISY	82 00	98 00

Table 4 7 LPC ITI TRAIN RS TEST RS

		TESTING	
		CLEAN	NOISY
TRAINING	CLEAN	98 26	49 06
	NOISY	12 66	98 00

Table 4 8 MEL ITI TRAIN RS TEST RS

		TESTING	
		CLEAN	NOISY
TRAINING	CLEAN	97 06	66 13
	NOISY	16 00	96 13

Table 4 9 SCALE ITI TRAIN RS TEST RS

		TESTING	
		CLEAN	NOISY
TRAINING	CLEAN	94 26	10 26
	NOISY	76 00	94 13

Table 4 10 LPC ITI, TRAIN RS TEST TS

		TESTING	
		CLEAN	NOISY
TRAINING	CLEAN	95 33	36 00
	NOISY	11 33	92 66

Table 4 11 MEL ITI TRAIN RS TEST TS

		TESTING	
		CLEAN	NOISY
TRAINING	CLEAN	94 93	59 86
	NOISY	13 86	91 86

Table 4 12 SCALE ITI TRAIN RS TEST TS

3 Experiment no 3 The performance of the recognition system as a function of signal to noise ratio using different front end processors is evaluated The training or generation of codewords is always done on the RS database and testing is done on both RS and TS databases Table 4 13 and Figure 4 2 shows the results when the recognizer is tested on RS database Table 4 14 and Figure shows the same for the TS database Under warp2 we have repeated the experiment only for clean speech and 15db SNR and the recognition rates obtained are 85 20 and 74 8 for RS database and 72 53 and 63 46 for TS database respectively This experiment mimics a practical speaker independent speech recognition environment It is generally difficult to know in advance the noise environment in which test utterance will be spoken Therefore the training is usually done under clean conditions and the features have to be robust to variations in environmental noise Further in practice the test utterance will be spoken by person other than trainer and the feature have to be robust to inter speaker variations

SNR	LPC	MEL SCALE	SCALE
CLEAN	98 00	98 26	97 06
30db	40 93	96 13	93 06
24db	16 00	86 93	89 73
18db	10 53	65 06	78 53
15db	10 13	49 06	66 13
12db	10 13	36 53	50 00
6db	10 00	17 33	29 33

Table 4 13 RECOGNITION RATE vs SNR TRAIN (CLEAN CONDITION RS) TEST RS

SNR	LPC	MEL SCALE	SCALE
CLEAN	94 26	95 33	94 93
30db	28 93	89 73	90 13
24db	14 13	76 53	84 40
18db	11 20	51 33	70 66
15db	10 26	36 00	59 86
12db	10 13	27 06	43 46
6db	10 00	15 33	26 66

Table 4 14 RECOGNITION RATE vs SNR TRAIN (CLEAN CONDITION RS) TEST TS

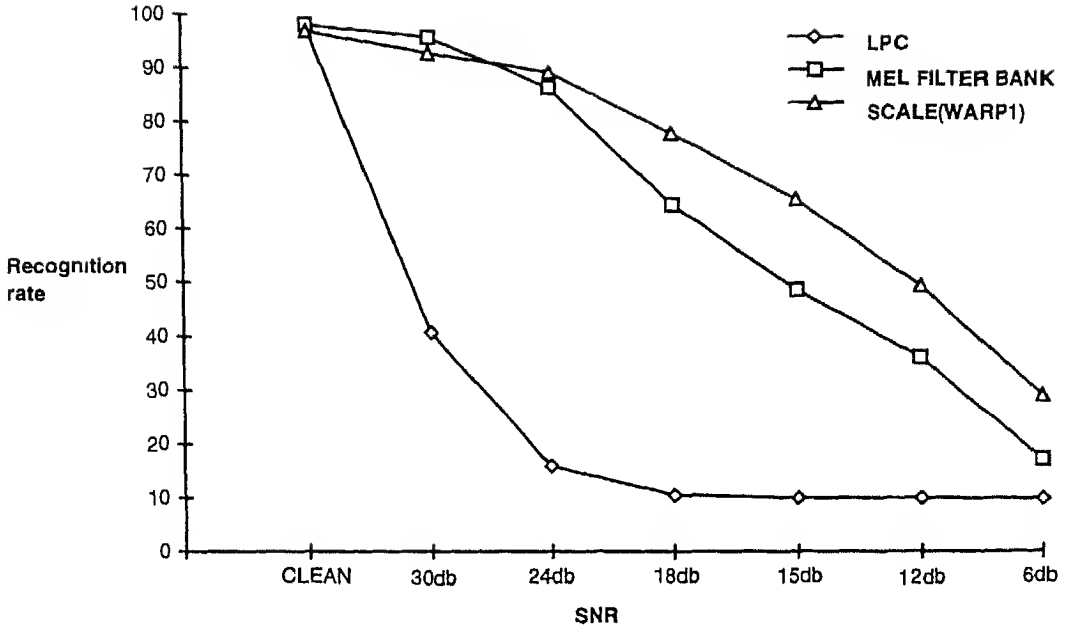


Figure 4 2 RECOGNITION RATE vs SNR TRAIN (CLEAN CONDITION RS) TEST RS

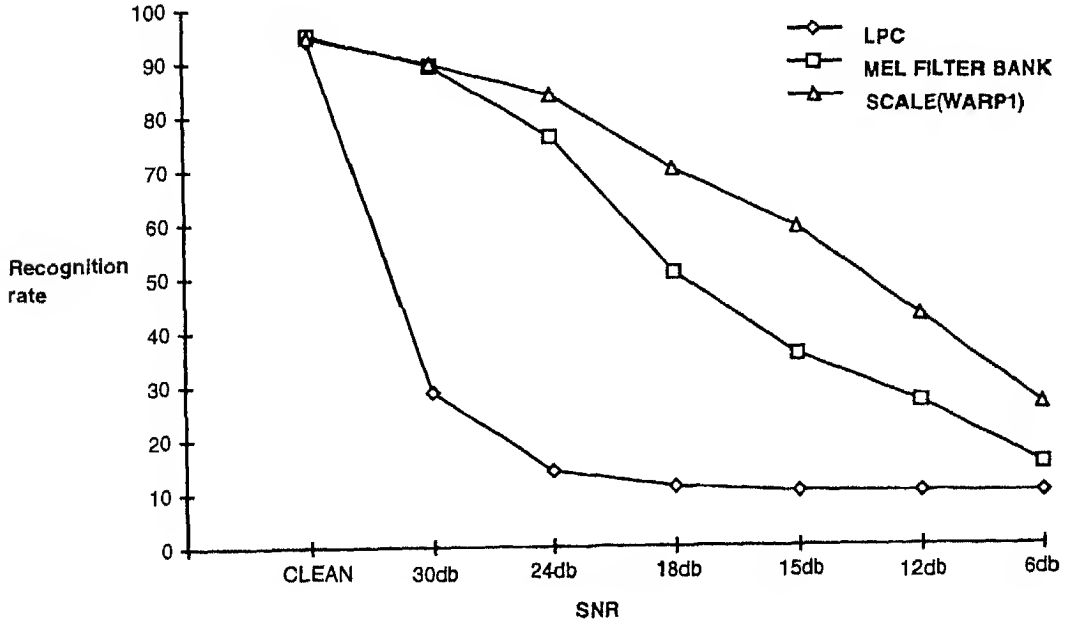


Figure 4 3 RECOGNITION RATE vs SNR TRAIN (CLEAN CONDITION RS) TEST TS

The results of Table 4 1 14 and Figure 4 2 3 show the following

1 The performance of the LPC front end processor falls much more rapidly with decreasing SNR as compared to the mel scale and scale cepstrum based front-end processor

2 The use of the transitional spectral information helps to improve the recognition performance significantly There is an improvement of approximately 5% in the recognition accuracy

3 Even at the signal to noise ratio of 15db (which resembles speech in a very noisy environment) the recognition system using scale cepstrum based front end processor significantly outperforms the mel scale and LPC based front end processor The difference is much more evident in the case of the TS database where the testing set is completely different from the training set which is the most necessary condition for a speaker independent recognition system

5 CONCLUSIONS AND FUTURE WORK

5 1 CONCLUSIONS

Although the linear prediction (LP) based processors are popular in commercial speech recognizers recent studies have shown improved performance using mel based features which have now almost become the industry standard A new front end processor based on scale transform based features have been proposed by Umesh *et al* [5] for speaker independent speech recognition In this thesis we have studied the performance of these three front end processors in terms of their classification performance for vowels and isolated digits when there is a mismatch between train and test conditions in terms of both the speakers and noise environments

From the tables in Chapter 3 we observe that for the practical case of noisy speech the scale cepstral features are comparable to mel cepstral features and significantly better than LPC features for vowel classification It can also be observed that Warp2 method of scale cepstrum performs the best among all the three methods

In Chapter 4 we compare the performance of an isolated digit recognizer based on vector quantization ideas using these three front end processors Experiment 3 of the chapter mimics a practical situation where we train under clean conditions and test under noisy conditions (since the noise environment under testing is not known in advance)

Under these conditions scale based features (both Warp1 and Warp2 methods) significantly outperform the mel based and LPC based features

5 2 FUTURE WORK

The results in this thesis indicate superior performance of scale cepstral features when compared to the conventional features The Warp2 method of scale cepstrum has only been recently experimented with and further studies need to be done to determine empirical rules such as the number of cepstral coefficients to be used in the feature vector the sampling rate to be used in the spectral domain and the type of liftering window if any to be used on the scale cepstral coefficients These empirical rules are necessary to optimize the tradeoffs between recognition accuracy and parsimony of feature space

BIBLIOGRAPHY

[1] L R Rabiner and B H Juang *Fundamentals of Speech Recognition* Prentice Hall 1993

[2] S Furui and M M Sondhi *Advances in speech signal processing* Marcel Dekker 1992

[3] S B Davis and P Mermelstein Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences IEEE Trans Acoustic Speech and Signal Processing vol ASSP 28 pp 357 366 Aug 1980

[4] C Jankowski H H Vo and R P Lippmann A comparison of signal processing front ends for automatic word recognition IEEE Trans Speech and Audio Processing vol 3 pp 286 293 July 1995

[5] S Umesh L Cohen N Marinovic and D Nelson Scale transform in speech analysis IEEE Trans Speech and Audio Processing Submitted 1996 Accepted Jan 1998

[6] S Umesh L Cohen N Marinovic and D Nelson Frequency warping in speech in Proc International Conference on Spoken Language Processing (Philadelphia USA) 1996

[7] S Umesh L Cohen and D Nelson Frequency warping and speaker normalization Proc ICASSP 97 (Munich Germany) pp 983 986 1997

[8] E Eide and H Gish A parametric approach to vocal tract length normalization Proc ICASSP 96 pp 346 349 1996

[9] L Cohen The scale representation IEEE Trans on Signal Processing vol ASSP 41 pp 3275 3292 Dec 1993

[10] H Wakita Normalization of vowels by vocal tract length and its application to vowel identification IEEE Trans on Acoustic Speech and Signal Processing vol ASSP 25 pp 183 192 April 1977

[11] L Cohen S Umesh N Marinovic and D Nelson Scale invariant speech analysis Proc of International Society for Optical Engineering (San Diego USA) 1995

[12] S Umesh L Cohen and D Nelson Frequency warping and psycho acoustic scales Proc of International Society for Optical Engineering (San Diego USA) 1996

[13] J G Wilpon L R Rabiner and A Bergh Speaker independent isolated word recognition using a 129 word airline vocabulary J Acoust Soc Am vol 72 No 2 Aug 1982

[14] S Furui Speaker independent isolated word recognition using dynamic features of speech spectrum IEEE Trans Acoustic Speech and Signal Processing vol ASSP 34 Feb 1986

[15] F K Soong and A E Rosenberg On the use of instantaneous and transitional spectral information in speaker recognition IEEE Trans Acoustic Speech and Signal Processing vol ASSP 36 June 1988

[16] L R Rabiner and Sambur An algorithm for determining end points of isolated utterances Bell Systems Tech Journal pp Feb 1975

[17] D Nelson Correlation Based Speech Formant Recovery Proc ICASSP 97 (Munich Germany) pp 1643 1646 1997

[18] S Umesh L Cohen and D Nelson Improved scale cepstral Analysis in Speech To appear in Proc IEEE ICASSP 98 (Seatle USA)